Lab 5 Explanation Document

By: Colin Ogrean

All of these measurements are in MS.

|  |  |  |  |
| --- | --- | --- | --- |
|  | qsort (C library) | unoptimized quicksort | optimized quicksort |
| lab5-gcc run 1 | 1666 | 997 | 2177 |
| lab5-gcc run 2 | 1672 | 991 | 2165 |
| lab5-gcc run 3 | 1678 | 995 | 2212 |
| lab5-clang run 1 | 1666 | 955 | 2261 |
| lab5-clang run 2 | 1668 | 952 | 2256 |
| lab5-clang run 3 | 1664 | 961 | 2313 |

Optimizations:

1. Branch-Free CMOV/Ternary Operations & 2. Branch-free XOR/Mask
2. //handle less than
3. int tmp\_lt = data[lt];
4. data[lt] = less ? data[i] : data[lt];
5. data[i] = less ? tmp\_lt : data[i];
6. lt += less;
7. i += less;
8. //handle greater than
9. int mask = -greater;
10. int t = (data[i] ^ data[gt]) & mask;
11. data[i] ^= t;
12. data[gt] ^= t;
13. gt -= greater;

3. Median of 3 pivot

//standard median-of-3 pivot selection

int median3\_index(int \*a, int lo, int hi)

{

int mid = lo + ((hi - lo) >> 1);

int x = a[lo], y = a[mid], z = a[hi];

if (x < y)

{

if (y < z)

return mid;

else if (x < z)

return hi;

else

return lo;

}

else

{

if (x < z)

return lo;

else if (y < z)

return hi;

else

return mid;

}

}

4. insertion-sort cutoff

static void insertion\_sort(int \*a, int n)

{

for (int i = 1; i < n; i++)

{

int x = a[i];

if (x >= a[i - 1])

continue;

int j = i - 1;

while (j >= 0 && a[j] > x)

{

a[j + 1] = a[j];

j--;

}

a[j + 1] = x;

}

}

Observations:

Clang consistently ran faster than gcc. Also, my timings seem all out of wack. I think this is because of a few reasons. The library-implementation being the slowest makes the most sense; using the “comparator”, it has to make SO MANY comparisons. The big weird thing is that my optimized version is so much slower than the unoptomized/first implementation.

This is likely because, despite lowering the branch mispredictions, I added so much other code that it still ended up consuming more cycles to do the same thing within the program. I know this isn’t what is supposed to happen (I think), so if theres actually something wrong with the code PLEASE LET ME KNOW because I’ve been stressing over this. Since my machine is ARM, I SSH’d into an AWS EC2 cloud instance running with a x86\_64 CPU to get better data, but to no avail. Sorry!

--Colin